CLAIM AMENDMENTS

Please amend the claims as follows:

1. (Original) A method of fabricating a reconfigurable processor for running moderately complex programming applications comprising:

(a) providing source code for a programming application,

(b) entering the source code in a control flow graph generating compiler to produce a control data flow graph of data flow control flow and branch points,

(c) extracting from the control flow graph basic blocks of code lying between branch points,

(d) from the code lying between the branch points generating intermediate data flow graphs,

(e) identifying clusters shared among dfgs at the highest level of granularity,

(f) from the identified clusters determine the largest common subgraph shared among the dfgs,

(g) scheduling the largest common subgraph for fast accomplishment of operations in the lcsg,

(h) applying the scheduled lcsg to the intermediate flow graphs replacing the unscheduled lcsg therein,

(i) scheduling the intermediate flow graphs containing the lcsg's for fast accomplishment of operations in the intermediate flow graphs to derive data patches having operations and timings of each intermediate flow graph,

(j) combining the data patches to include operations and timing of the lcsg with operations and timings of each intermediate subgraph that are outside the lcsg,

(k) from the combined data patches scheduling for process time reduction multiple uses of the lcsg operations and timings necessary to accomplish operations and timings of all intermediate subgraph employing the lcsg , and

(l) implementing in hardware having mixed granularities the operations and timing of the lcsg including:

(i) partitioning,

(ii) placing, and

(iii) interconnection routing.

2. (Original) In a method of making an integrated circuit for use as a hardware implemented part of a programmed operation implemented in software and hardware; the improvement comprising identifying hardware circuit elements for execution of a largest common subgraph common among a set of flow graphs representing the programmed operation; partitioning into blocks the circuit elements; arranging the blocks on an area representative of an available area of a surface of a substrate on which the circuit elements are to be formed; routing interconnections among the blocks; partitioning into sub-blocks the circuit elements of each block; arranging each sub-block on an area representative of the block from which it has been partitioned, routing interconnections among the sub-blocks and iteratively partitioning and routing among lesser sub-blocks until the individual circuit elements have been placed and routed.

3. (Original) The method according to claim 2, wherein the steps of routing comprise locating conductors and switches for interconnections among blocks, sub-blocks and circuit elements.

4. (Original) The method according to claim 3, wherein locating conductors and switches further comprises locating variable switches to effect variable conductive paths among the blocks, sub-blocks and circuit elements.

5. (Original) A method of scheduling process elements of hardware implementing a program operation, comprising:

(a) developing a control data flow graph from the software;

(b) using a first, non-exhaustive scheduling algorithm to relatively quickly arrive at a first scheduling of the process elements;

(c) using a second more exhaustive scheduling algorithm for at least one and less than all selected paths of the control data flow graph to reduce the time of execution thereof; and

(d) once all paths of the control data flow graph have been scheduled, including all of the second more exhaustive scheduling, merging all of schedules, respecting data and resource dependencies.

6. (Original) The method of scheduling according to claim 5, wherein step (b) comprises PCP scheduling.

7. (Original) The method of scheduling according to either claim 5 or 6, wherein step (c) comprises branch and bound based scheduling.

8. (Original) A dedicated integrated circuit for performing the program operation having processing elements scheduled according to claim 5.

9. (Original) A dedicated integrated circuit for performing the program operation having processing elements scheduled according to claim 6.

10. (Original) A dedicated integrated circuit for performing the program operation having processing elements scheduled according to claim 7.

11. (Original) The method of forming an application specific reconfigurable circuit, comprising:

(a) providing source code for an application to be run by the circuit,

(b) deriving flow graphs representing separate portions of the application,

(c) identifying at least one largest common flow graph from at least two of the separate portions of the application; and

(d) in hardware, configuring circuitry to be shared by the separate portions of the application.

12. (New) A method of fabricating an integrated circuit implementing multiple program operations comprising:

(a) providing source code for the multiple program operations;

(b) deriving control flow graphs for selected multiple program operations;

(c) identifying basic blocks of the control flow graphs;

(d) developing data flow graphs of at least a plurality of the basic blocks;

(e) identifying a common subgraph shared by at least a pair of the basic blocks of control flow graphs of the separate program operations;

(f) scheduling that makes up the functions(+, -, *) common subgraph to quicken shared process occurrences represented by the common subgraph;

(g) scheduling the quickened shared processes of the common subgraph for operation in each of the multiple program operations;

(h) overall scheduling of processing units to carry out the common subgraph by:

(i) clustering the processes of the processing units of the common subgraph into a macroblock having nodes representing the processes of common subgraph and at least a plurality of unconditional, conditional and reconfiguration edges running between nodes;

(ii) determining the relative delay among the possible paths through the common subgraph;

(iii) performing branch and bound scheduling for at least the longest delay time path and less than all paths through the common subgraph; and

(iv) merging all of the schedules;

(i) laying out the arrangement of circuit elements for implementation of the integrated circuit in hardware including;

(i) grouping the circuit elements into first level clusters; and

(ii) placing the first level clusters by grouping the first level clusters together to form second level clusters and placing the second level clusters.

13. (New) The method of fabricating an integrated circuit according to claim 12, wherein step (e) comprises identifying seed basic blocks by identifying candidate seed basic blocks among the identified basic blocks of the at least a plurality of control flow graphs, and comparing candidate seed basic blocks from control flow graphs of separate program operations.

14. (New) The method of fabricating an integrated circuit according to claim 13, wherein identifying seed basic blocks comprises identifying basic blocks that lie inside a loop.

15. (New) The method of fabricating an integrated circuit according to claim 14, wherein identifying basic blocks that lie inside a loop comprises identifying one of:

(i) a single nested level loop with only one basic block;

(ii) a single nested level loop with more than one basic block; and

(iii) a multi-level nested loop.

16. (New) The method of fabricating an integrated circuit according to claim 14, wherein identifying basic blocks that lie inside a loop comprises identifying one of:

(i) a single nested level loop with more than one basic block; and

(ii) a multi-level nested loop.

17. (New) The method of fabricating an integrated circuit according to claim 16, wherein identifying seed basic blocks further comprises identifying basic blocks of control flow graphs of separate program operations under like control.

18. (New) The method of fabricating an integrated circuit according to claim 17, wherein identifying seed basic blocks further comprises determining a count of each operation type in a basic block of similar class of decision, merge or pass.

19. (New) The method of fabricating an integrated circuit according to claim 18, wherein identifying seed basic blocks further comprises examining edges in a data flow graph of candidate seed basic blocks of control flow graphs from the separate programming operations.

20. (New) The method of fabricating an integrated circuit according to claim 19, wherein examining edges comprises classifying edges in the data flow graphs on the bases of source and destination node operation type.

21. (New) The method of fabricating an integrated circuit according to claim 20, wherein examining edges includes eliminating edges of one data flow graph having a source operation to destination operation not found in the other data flow graph the edges of which are being examined.

22. (New) The method of fabricating an integrated circuit according to claim 21, further comprising accomplishing the edges eliminated in other than application specific integrated circuit (ASIC).

23. (New) The method of fabricating an integrated circuit according to claim 22, wherein accomplishing the edges eliminated in other than ASIC comprises accomplishing the edges eliminated with look up tables (LUTs).

24. (New) The method of fabricating an integrated circuit according to claim 20, wherein examining edges further comprises comparing associativity among edges being compared.

25. (New) The method of fabricating an integrated circuit according to claim 24, wherein comparing associativity comprises determining numbers of predecessor, siblings, companions and successors of edges being compared.

26. (New) The method of fabricating an integrated circuit according to claim 12, wherein step (f) comprises ASAP scheduling the common subgraph.

27. (New) The method of fabricating an integrated circuit according to claim 12 or 23, further comprising providing the common operations in the common subgraph in an application specific integrated circuit (ASIC).

28. (New) The method of fabricating an integrated circuit according to claim 12, further comprising identifying at least one further common subgraph shared by the at least a pair of the basic blocks and operating on the at least one further common subgraph pursuant to steps (f) - (i).

29. (New) The method of fabricating an integrated circuit according to claim 12, wherein step (e) comprises identifying the largest common subgraph shared by the at least a pair of the basic blocks.

30. (New) The method of fabricating an integrated circuit according to claim 12, wherein step (f) comprises providing switching of differing delays among processes of the common subgraph to effect the subgraphs operating each of the multiple program operations.

31. (New) The method of fabricating an integrated circuit according to claim 30, wherein providing switching comprises providing multiplexers operative to switch in alternative delays between processes of the common subgraph.

32. (New) An integrated circuit fabricated by the method of claim 12.

33. (New) Computer programming having routines implementing the method of claim 12.

34. (New) In a method of fabricating a reconfigurable integrated circuit including developing a data flow graph for at least a portion of the operations of the integrated circuit; the improvement comprising:

(a) scheduling the at least a portion of the operations by calculating the delay along each path through the data flow graph from a processing element being scheduled to a sink node of the data flow graph including:

(i) adding to edges of the data flow graph reconfiguration edges representing reconfiguration of that part of the integrated circuit effecting the at least a portion of the operations; and

(ii) including in the calculation of delay along each path the effect of reconfiguration on delay time; and

(b) scheduling first, as the longest in duration path, the longest of processing times of processing elements including reconfiguration delay and the calculated delay of step (a).

35. (New) A method of fabricating a reconfigurable integrated circuit according to claim 34, further comprising scheduling all shorter in duration paths within the time established for the longest in duration path.

36. (New) An integrated circuit fabricated by the method of claim 34.

37. (New) Computer programming having routines implementing the method of claim 34.

38. (New) In a method of fabricating a reconfigurable integrated circuit including developing a data flow graph for at least a portion of the operations of the integrated circuit; the improvement comprising:

(a) scheduling the at least a portion of the operations by calculating the delay along each partial critical path through the data flow graph from a processing element being scheduled to a sink node of the data flow graph;

(b) calculating with a second, more exhaustive calculation the delay through the path determined to be the longest in duration in step (a); and

(c) determining whether the calculation of step (b) confirms the calculation of the longest-in-duration path calculation of step (a);

(i) scheduling first the longest-in-duration path calculation in step (a) if step (b) confirms step (a); or

(ii) determining from among the remaining paths the longest-in-duration path other than the path so determined in step (a) and scheduling first the longest-in-duration path from among the remaining paths.

39. (New) The method of fabricating a reconfigurable integrated circuit according to claim 38, wherein step (c) (ii) comprises calculating with the second, more exhaustive calculation the delay through the longest-in-duration path from among the remaining paths.

40. (New) The method of fabricating a reconfigurable integrated circuit according to claim 38 or 39, wherein the more exhaustive calculation comprises branch and bound calculation.

41. (New) The method of fabricating a reconfigurable integrated circuit according to claim 38 or 39, further comprising scheduling all shorter in duration paths within the time established for the longest in duration path.

42. (New) An integrated circuit fabricated by the method of claim 38.

43. (New) A computer programming having routines implementing the method of claim 38.

44. In a method of fabricating a reconfigurable integrated circuit including developing a data flow graph for at least a portion of the operations of the integrated circuit; the improvement comprising:

(a) scheduling an operation of a processing element in a loop;

(b) scheduling one or more buffer times following the operation of the processing element in a loop;

(c) scheduling an operation of all further processing elements dependent on the processing element in a loop beginning after the buffer time to permit communication to the further, dependent elements of the status of the processing element in a loop.

45. (New) The method of fabricating a reconfigurable integrated circuit according to claim 44, wherein step (a) comprises scheduling an estimated most probably number of iterations of the operation of the processing element in a loop.

46. (New) The method of fabricating a reconfigurable integrated circuit according to claim 44, wherein step (a) comprises scheduling a single iteration of the operation of the processing element in a loop.

47. (New) The method of fabricating a reconfigurable integrated circuit according to claim 44, further comprising providing for notification of delay of all dependent elements during buffer times upon the processing element in the loop iteratively operating in excess of the scheduled operation of step (a).

48. (New) The method of fabricating a reconfigurable integrated circuit according to claim 47, further comprising providing for delay of a network schedule manager and a logic schedule manager upon the processing element in the loop iteratively operating in excess of the scheduled operation of step (a).

49. (New) An integrated circuit fabricated by the method of claim 44.

50. (New) Computer programming having routines implementing the method of claim 44.

51. (New) In a method of fabricating a reconfigurable integrated circuit including developing a data flow graph for at least a portion of the operations of the integrated circuit; the improvement comprising:

(a) providing a first loop having a first processing element;

(b) providing an output of the first loop to a second loop having a second processing element;

(c) providing an output of the second loop to an input of the first loop;

(d) scheduling an identical number of operations of the first and second processive elements;

(e) scheduling an operation of all further processing elements dependent on the first and second loops beginning after a buffer time to permit communication of the status of first and second loops to the further, dependent elements.

52. (New) The method of fabricating a reconfigurable integrated circuit according to claim 51, wherein step (a) comprises scheduling an estimated most probably number of iterations of the operation of the first and second loops.

53. (New) The method of fabricating a reconfigurable integrated circuit according to claim 51, wherein step (a) comprises scheduling a single iteration of the operation of the first and second loops.

54. (New) The method of fabricating a reconfigurable integrated circuit according to claim 51, further comprising providing for notification of delay to all dependent elements during buffer times when the first and second loops iteratively operate in excess of the scheduled operation of step (a).

55. (New) The method of fabricating a reconfigurable integrated circuit according to claim 54, further comprising providing for delay of a network schedule manager and a logic schedule manager when the first and second loops iteratively operate in excess of the scheduled operation.

56. (New) In a method of fabricating a reconfigurable integrated circuit including developing a data flow graph for at least a portion of the operations of the integrated circuit; the improvement comprising:

(a) providing a loop having control nodes within the loop;

(b) scheduling the loop by:

(i) scheduling the longest in duration path through the loop;

(c) scheduling an operation of all further processing elements dependent on the loop after a buffer time to permit communication of the status of the loop to the further, dependent elements.

57. (New) The method of fabricating a reconfigurable integrated circuit according to claim 56, wherein step (a) comprises scheduling an estimated most probable number of iterations of the operation of the loop, each iteration employing the longest-in-duration path through the loop.

58. (New) The method of fabricating a reconfigurable integrated circuit according to claim 56, wherein step (a) comprises scheduling a single iteration of the loop employing the longest-in-duration path through the loop.

59. (New) The method of fabricating a reconfigurable integrated circuit according to claim 56, further comprising providing for notification of delay to all dependent elements during buffer times when the loop iteratively operates in excess of the scheduled operation of step (a).

60. (New) The method of fabricating a reconfigurable integrated circuit according to claim 59, further comprising providing for delay of a network schedule manager and a logic schedule manager when the loop iteratively operates in excess of the scheduled operation of step (a).

61. (New) An integrated circuit fabricated by the method of claim 56.

62. (New) Computer programming having routines implementing the method of claim 56.